The Influence of Spatial and Transient Circuit Variations on Energy and Accuracy in Stochastic Computing Circuits

Bert Moons
Department of Electrical Engineering - ESAT
KU Leuven

Marian Verhelst
Department of Electrical Engineering - ESAT
KU Leuven

***Abstract*—The continued scaling of feature sizes in integrated circuit technology leads to an increase of uncertainty and unreliability in circuit behaviour. Maintaining the paradigm of deterministic Boolean computing therefore becomes increasingly challenging. Stochastic computing (SC) processes digital data in the form of long pseudo-random bit-streams denoting probabilities. Its probabilistic aspect makes it less vulnerable to errors and uncertainty. This suggests SC is a possible alternative for classic binary digital systems when very high circuit variations are present. This paper investigates quantitatively the low level impact of different noise/variation sources on the accuracy and energy consumption of a basic SC multiplier. This performance is compared to a classic binary multiplier, subjected to the same variability. The comparison is made with circuit variation models, extrapolated from 40nm CMOS technology simulation results, to estimate the impact of further scaling. Our analysis shows SC only is an interesting alternative to binary in technologies with low energy per bit-operation and significant transient circuit variations.**

***Index Terms*—Stochastic Computing, variability**

I. Introduction

Digital electronics has always relied on error-less circuit operation. Precise Boolean functionality, defined in a deterministic logical layer is translated into a physical layer that produces voltages. These can be interpreted as the needed exact logic values. This abstraction has been successful, but becomes ever more costly. All forms of noise and uncertainty in the physical layer have to be compensated for through more complex and energy-hungry designs. Recently, new research is focussing on novel ways to handle device uncertainty in a more efficient way. A very promising class of techniques, labeled "Stochastic Computation", exploits probability theory to deal with variations. Shanbhag et al. give an overview of different techniques [1]. Stochastic Computing (SC), a promising computational technique introduced by Gaines [2] processes data in the form of digitized probabilities. SC has three main advantages over conventional computing approaches. First it uses very low complexity building blocks, making it suitable for massively parallel processing. Second, SCs probabilistic aspect makes it inherently tolerant to soft transient errors (such as bit-flips) and robust against spatial variations. A third advantage is the potential to create logic with scalable precision. Shortened bit-streams can provide an early estimate of a number value. This concept provides an easy possibility to trade-off precision for energy, an advantage that can be well exploited in ultra-low energy electronics for e.g. wearable or multimedia applications. Due to SC's error tolerance, the logic type seems a good alternative for digital designs in technologies suffering from high uncertainty. Although SC has been known for decades, very few physical implementations have been made. Recently SC has been used in LDPC decoding [3] and in basic image processing systems [4] [5]. Alaghi and Hayes [6] and Qian an Riedel [7] [8] have proposed synthesis approaches for classes of combinational circuits, hereby enabling a formal approach to generate complex and in some cases reconfigurable arithmetic functions. Most previous research, however, has been on the mathematical/system level and does not take real circuit variations into account. We focus directly on the impact of several types of device variations on the output precision in SC. This work is also the first to focus directly on the energy usage of SC. As a benchmarking and comparison circuit we choose a SC and a binary multiplier. This paper is organized as follows. Section II gives an overview of stochastic numbers and SC arithmetic blocks. Section III discusses different sources of circuit variations and its impact on SC and binary multipliers. Section IV concludes this work.

II. Stochastic Computing

Stochastic numbers (SN) are bit-streams containing N1 $1's$ and N0 $0's$ denoting the unipolar (UP) number p = N1/(N1+N0). Since p will always lie in the real-number interval [0,1], it can be interpreted as the probability that the bit-stream outputs a 1. A bipolar (BP) interpretation of the bit-stream is possible by transforming p onto the [-1,1] interval (s=2p-1). The precision of the stochastic number is determined by the length of the bit-stream. A bit-stream of (L=256=2^8) bits has a maximal theoretical accuracy of 8 binary bits. A typical SC system exists out of a binary-to-stochastic (BTS) conversion unit, stochastic arithmetic and a stochastic-to-binary (STB) converter (figure 1). The BTS unit can be easily implemented using LFSR pseudo random number generators [9]. These can be proven to generate nearly exact approximations of the wanted binary input value. For conversion from SC to binary a simple binary counter suffices. The used stochastic arithmetic gate depends on the number interpretation. Multiplication can be done by using an AND-gate in the UP format (figure 2), or an XNOR in the BP format. Scaled addition can be implemented using a MUX-gate in both cases [9]. The INV-gate implements (1-p) in the UP and (-p) in the BP format. More complex gates such as comparators and linear gain functions are nontrivial in SC (in contrast to binary logic) and can be implemented using the synthesis approaches from [6] and [7] or by using an FSM-based system [10] . The further analysis will be done on a stochastic multiplier in the UP format.

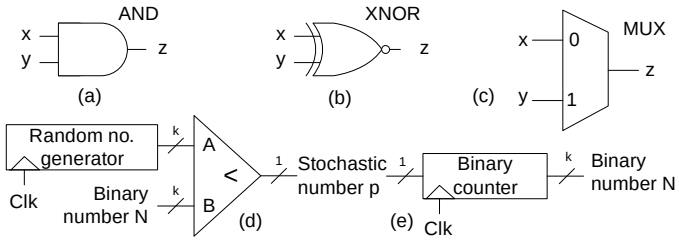


Fig. 1. Examples of basic SC arithmetic gates. (a) Unipolar multiplier (b) Bipolar multiplier (c) Scaled adder (d) Binary-to-stochastic converter (e) Stochastic-to-binary converter.

III. Impact of Inherent, Spatial and Transient variation on Accuracy and Energy in Digital Systems

A. Accuracy

There are three major sources of errors in advanced technology digital computations: errors inherent to the used logic type (type I), errors due to spatial circuit variations (type II) and errors due to transient circuit variations (type III). Examples of type I errors are quantization faults in the binary logic type, and faults due to correlation in the SC logic type. In binary systems the quantization noise power or squared root-mean-square error (RMSE) equals:

$$\sigma^2_{binary} = RMSE^2 = \frac{\delta^2}{12} = \frac{1}{12 \cdot 2^{2n}} \tag{1}$$

where n is the binary precision. Observe that the variance drops quadratically with 2^n. Inherent system noise in SC can be estimated by observing that the interpreted value p of a randomized bit-stream is approximately binomially distributed. Even if near exact LFSR stochastic number generators are used, correlation effects randomize the stochastic number after a few (> 2) logic stages. Better estimations exist [11], yet the accuracy of the binomial assumption is sufficient for our study. If the number value is uniformly distributed over its full interval [0,1], the mean variance is determined by:

$$\sigma^2_{SC} = RMSE^2 = \int_0^1 \frac{p \cdot (1-p)}{L} \, \mathrm{d}p = \frac{1}{6 \cdot L} \tag{2}$$

where L is the length of the bit-stream. The inherent variance of SC only drops linearly with the bit-stream length L. By comparing equations (1) and (2), it becomes clear that very long bit-streams are needed to achieve high precision.

$$\sigma^2_{SC} = \sigma^2_{binary} \iff L = 2^{2n+1} \tag{3}$$

For example: in order to achieve the same inherent noise power of an 8 bit binary system, a 131072 long stochastic stream is needed. Type II errors stem from spatial circuit variations. These are variations that are random in space, but fixed in time, such as random doping fluctuations or any kind of inter- or intra-die variations. These are already omnipresent in current transistor technologies and will become more important in future technologies. Since the critical path of a SC multiplier is fixed and very short (in contrast to the critical path of a binary multiplier), it is expected that the influence of spatial variations on SC performance is limited (see III-D). To accomodate type III errors, we simulate fast transient circuit variations. These are variations that are random in time and space, such as random bit-flips, radiation effects, or supply-voltage ringing. In current technologies spatial variations are still the dominant source of uncertainty, but transient variations might become more important in more advanced CMOS or in post-Si technologies when dopant levels and voltage headroom further decreases. SC's probabilistic aspect makes it less vulnerable to this type of variations than binary systems (see III-D).

B. Energy

For a fair comparison between SC and binary multipliers, the required energy for the arithmetic function should be compared for an identical RMSE at the output. In a SC system the energy consumption can be summarized as.

$$E_{SC} = \alpha \cdot C \cdot V_{dd}^2 \cdot f_{SC} \tag{4}$$
$$f_{SC} = f_{bin} \cdot \frac{L}{P}$$
$$V_{dd} = f(f_{SC}, critical\,path)$$

Where α is the circuit activity, C is the technology capacitance, V_{dd} is the supply voltage, f_{SC} and f_{bin} are the SC and binary clock frequencies, L is the bit-stream length and P is the used degree of parallelization. A SC system uses long bit-streams to achieve high accuracy and thus requires high clock-speeds to reach a given computing delay. However, the usage of very short data paths will enable using lower supply voltages for a given delay. To reduce energy and total delay, arithmetic functions can be parallelized with the factor P. This can be done with little overhead due to the low gate complexity. From equations (4) the energy used for stochastic (eq. 5) and binary (eq. 6) multiplication can be estimated as :

$$E_{SC} = k \cdot L \tag{5}$$

$$E_{binary} = c \cdot (2^n)^{\frac{1}{2}} \tag{6}$$

where L is the stream-length, k and c are determined by supply voltage, clock speed and technology. k is the mean energy per bit-operation in SC, c can be interpreted equivalently. By combining equations (1) and (2) with respectively equations (5) and (6) the following relations can be found:

$$E_{SC} = \frac{k}{6 \cdot RMSE^2} \tag{7}$$

$$E_{binary} = c \cdot (\frac{1}{\sqrt{12} \cdot RMSE})^{\frac{1}{2}} \tag{8}$$

To achieve lower energy in SC than in binary at a given RMSE, the constant k has to be small compared to c. How k and c vary with spatial variations can be simulated (section III-C and III-D).

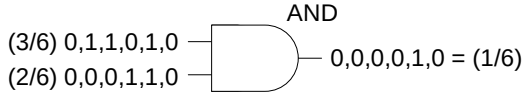
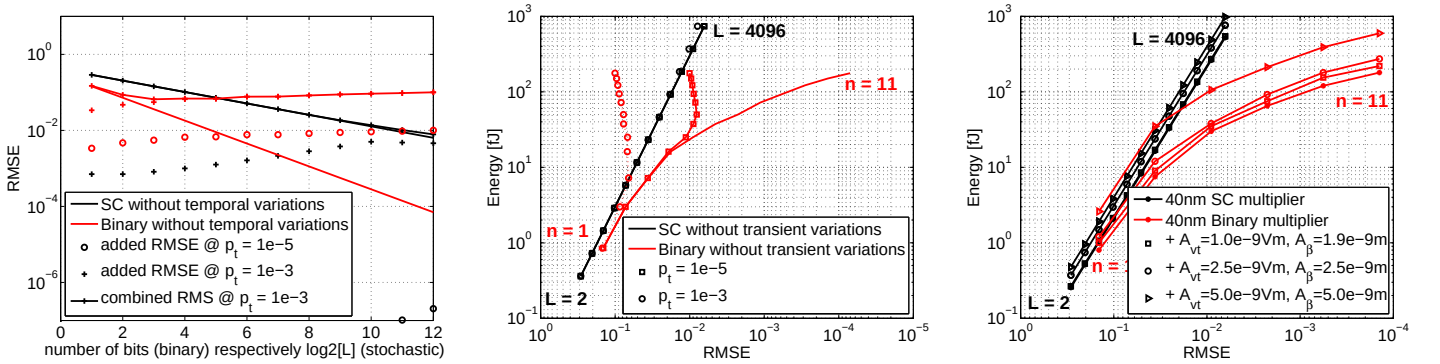


Fig. 2. Example of SC multiplication. Two input bit-streams (3/6) and (2/6) are AND-ed. Input correlations lead to errors.

C. Simulation setup

In order to quantitatively compare the accuracy in SC and binary digital electronics, we have simulated a SC multiplier as well as a standard carry-select multiplier (without pipelining). Both systems are equally exposed to the previously mentioned sources of uncertainty. Both type I (inherent) and type III (transient) errors can be simulated on the system level. Type II (spatial) variations require transistor level simulations. Circuit simulations for SC are set-up as follows: two random bit-streams p_a and p_b are multiplied using a 40nm AND-gate. At a given clock frequency, supply voltage is swept. For every voltage step the accuracy impact due to spatial variations is recorded. This dependency is only a function of the used circuit, spatial variations and frequency. The minimal supply voltage at which no type II errors occur is used to further assess the impact of type I an III errors. The binary multiplier works at a f_{bin} of 31MHz (period = 32ns). The SC-multiplier clocks at a much higher f_{SC} of 496 MHz (period = 2ns) at a parallelization degree of $P = 16$. All SC and binary circuit simulations are done using 15 Monte-Carlo runs, which offer sufficient resolution for the targeted first order analysis. To mimic more advanced technologies with more uncertainty, extra Vt- and β-mismatch is added using a verilog-A behavioural model. Transient circuit variations such as bit-flips or cycle-to-cycle variations can be modelled by using an XOR-gate on every logical output node. Input stream p_a will be distorted at a rate p_t, where p_t will be very low. The resulting p_{out} equals xor(p_a,p_t). If a bit of bit-stream p_t equals 1, the corresponding bit of stream p_a will invert.



(a) Transient variations + inherent RMSE

(b) Transient variations+ inherent RMSE

(c) Spatial variations + inherent RMSE, A_{vt} and A_β are pelgrom's constants

Fig. 3. Simulation results

D. Simulation results

Figure 3 shows the results of our simulations. Figure 3(a) plots the additive and combined RMSE against the bitwidth n for binary systems and against the stream length L for SC systems, and this for various transient error rates p_t. By introducing transient errors, the achieved RMSE will be higher for the same n or L. This figure clearly shows that SC multipliers can reach much lower RMSE under the same circumstances, by using longer bit-streams. Figure 3(b) further illustrates this by plotting the energy consumption in function of RMSE for binary and stochastic implementations. Even at the relatively low transient error rate of $1e-5$, it is impossible to achieve an RMSE lower than $3e-8$ using a binary system. This corresponds to a binary accuracy of 5 bits. At a p_t of $1e-3$, only 2 bit binary accuracy can be reached. The performance degrades further when using larger bit-widths. This degradation is due to two reasons. First, the number of logical/flippable nodes increases quadratically in a binary carry-save multiplier. Second, MSB-nodes flip at the same rate as LSB-nodes, but contribute much more to the global RMSE. SC clearly has an advantage over binary computing in the case of transient circuit variations. The contribution of type III variations to the global RMSE at a flip rate of $1e-5$ is negligible. SC's reduced hardware complexity leads to less logical nodes and flipped bits always lead to an LSB error.

Figure 3(c) plots the energy of multiplication using both the SC and the binary logic type, as a function of achieved RMSE for different amounts of type II variations. These simulation results take type I variations into account and reasonably coincide with the predictions of equations (7) and (8). For SC in the 40nm case, k equals $0.13fJ/bit$. In the case with highest extra variations, $k = 0.24fj/bit$. Ignoring the $n = 1$ data-point, the best fit for the 40nm binary case gives $c = 3.3$. In the case with highest variations $c = 15.5$. SC has no advantage over binary for any RMSE in the 40nm case and only down to $3e-2$ RMSE (3 bits binary precision) in the high variations case. It is clear that SC only outperforms binary multiplication in terms of energy usage when very high RMSE are tolerated and high spatial variations are present. High RMSE can be allowed in some image processing applications, such as edge detection [4]. At high RMSE, the energy usage is generally similar between the two systems, but rises quicker in the binary multiplier than in SC with increasing spatial variations. However, due to the quadratic dependence of RMSE in SC, binary logic still performs better. Furthermore, the energy usage in binary can be reduced by pipelining the multiplier, this effectively reduces the impact of type II variations on delay and energy. Pipelining is not possible in the SC multiplier, since it only has a single stage.

The usage of the proposed framework allows to quickly evaluate the performance of SC in a new technology if parameters k, c and p_t are known. In technologies with sufficiently low k and high p_t, SC will be preferable to binary computation.

IV. Conclusion

By comparing the minimum energy per operation needed to implement digital multiplication in a stochastic and in a binary way, we can evaluate the performance of SC. Inherent quantization and correlation noise is much larger in SC than in binary logic. In 40nm, SC is less energy-efficient than binary. If very high spatial variations are present, SC outperforms binary up to 3-4 bit binary precision, which is no significant performance improvement. However, when transient circuit variations are present, SC greatly outperforms binary logic. At a flip rate of $1e-5$, it is impossible for binary systems to perform better than a minimal RMSE of $8e-3$. SC is tolerant to transient variations and can reach a lower RMSE by using longer bit-streams. In conclusion, our proposed framework indicates that SC is a good alternative to binary only for technologies with a low k (energy per bit-operation) that suffer from significant transient circuit variations.

References

[1] N. Shanbhag, "Stochastic computation," *DAC*, 2010.

[2] R. Gaines, "Stochastic computing systems," *Advances in information systems science*, 1969.

[3] A. Naderi, S. Mannor, M. Sawan, and W. Gross, "Delayed stochastic decoding of ldpc codes," *IEEE tran. Signal Proc.*, 2011.

[4] A. Alaghi, C. Li, and J. Hayes, "Stochastic circuits for real-time image processing applications," *Design automation conference (DAC)*, 2013.

[5] P. Li and D. Lilja, "Using stochastic computing to implement digital image processing," *ICCD*, 2011.

[6] A. Alaghi and J. Hayes, "A spectral transform approach to stochastic circuits," *International conference on computer design (ICCD)*, 2012.

[7] W. Qian and M. Riedel, "The synthesis of robust polynomial arithmetic with stochastic logic," *Design Automation Conference (DAC)*, 2008.

[8] W. Qian, X. Li, and R. Marc, "An architecture for fault-tolerant computation with stochastic logic," *IEEE transactions on computers*, 2011.

[9] A. Alaghi and J. Hayes, "Survey of stochastic computing," *ACM transactions on embedded computing*, 2012.

[10] B. Brown and H. Card, "Stochastic neural computation i : computational elements," *IEEE transactions on computers*, 2001.